Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-91-27

ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge

A. Bernardi, H. Boley, Ph. Hanschke,
K. Hinkelmann, Ch. Klauck, O. Kühn,
R. Legleitner, M. Meyer, M. M. Richter,
F. Schmalhofer, G. Schmidt, W. Sommer

September 1991

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum
für
Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Philips, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❏ Intelligent Engineering Systems
- ❏ Intelligent User Interfaces
- ❏ Intelligent Communication Networks
- ❏ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge

A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer

DFKI-RR-91-27

A short version of this paper will be published in the Proceedings of Avignon 91, Expert Sytems and their Applications: Tools, techniques and Methods

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-8902 C4).

ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge

Ansgar Bernardi, Harold Boley, Philipp Hanschke, Knut Hinkelmann,
Christoph Klauck, Otto Kühn, Ralf Legleitner, Manfred Meyer, Michael M. Richter,
Franz Schmalhofer, Gabi Schmidt, Walter Sommer

DFKI GmbH, Postfach 2080, 6750 Kaiserslautern, Germany
Phone: +49-631-205-3470,
Fax: +49-631-205-3210
email: mmr@dfki.uni-kl.de

Abstract

A global description of an expert system shell for the domain of mechanical engineering is presented. The ARC-TEC project constitutes an AI approach to realize the CIM idea. Along with conceptual solutions, it provides a continuous sequence of software tools for the acquisition, representation and compilation of technical knowledge. The shell combines the KADS knowledge-acquisition methodology, the KL-ONE representation theory and the WAM compilation technology. For its evaluation a prototypical expert system for production planning is developed. A central part of the system is a knowledge base formalizing the relevant aspects of common sense in mechanical engineering. Thus, ARC-TEC is less general than the CYC project but broader than specific expert systems for planning or diagnosis.

Keywords: CIM, common sense, constraints, CYC, feature descriptions, forward/backward rules, integrated product model, KADS, KL-ONE, knowledge compilation, mechanical engineering, model-based knowledge acquisition, production planning, skeletal plans, terminological representation languages, WAM

Table of Contents

1 Introduction

One particular example of an *intelligent specialist* is a mechanical engineer. Even if his domain consists of highly specialized artifacts, he uses all kinds of human knowledge and experience. He knows data, tools and methods, their advantages and when to use them. His knowledge is embedded in broader contexts and in everyday life and his expert knowledge is supported by common sense. Although the possibility of using AI techniques in CAD has been known for a long time [Lat78], only in recent years the necessity of formalizing common sense has been generally recognized. The most ambitious approach to represent common sense is probably D.Lenat's CYC project at MCC in Austin/Texas [Len90].

The ARC-TEC project (which started in the second half of 1989) is much more modest but has some resemblance to the CYC project. While in CYC the target is all knowledge that everybody has, ARC-TEC is aiming at the knowledge that enables a mechanical engineer to carry out his tasks. The common sense is restricted to those aspects which play a role in a technical world. On the other hand, in this engineering world much deeper and more specialized knowledge has to complement the common sense knowledge.

There are expert systems as well as traditional programs which can solve some mechanical engineering problems. Their main weak point is the isolated character and this deficit has been the focus of attention of many activities which run under the headline “Computer Integrated Manufacturing”. If there are knowledge islands the natural thing seems to be to build bridges; if these islands are programs such bridges are interfaces. Interfaces, however, are only possible, if the contents in the different modules can be transformed into each other in a sufficiently easy way. If the information in an island can only be understood on the basis of a complex background, then the background has to be represented and the transformation has to refer to it. For different expert systems, even in the same area of application, the represented knowledge is usually not sufficient to bridge the gap between them. The systems may deal with the same kind of problem but they will view things in a different and often incompatible manner. To understand the other system one needs not only some syntactic translator but a whole background of general knowledge and experience. The ARC-TEC approach may be seen as providing this background for (selected subdomains of) mechanical engineering; one can regard it as an AI approach to realize the CIM idea. Through acquisition the knowledge available to human experts is made accessible to a formal representation so that it can be further transformed and processed by compilation.

2 General Structure

Expert system development takes place on three levels:

- The cognitive level where the problems are discussed in an informal though concise way.
- The representation level where matters are represented formally.
- The implementation level where data structures and available programming languages are used.

For a thorough investigation ARC-TEC deals with all three levels. They are connected as follows:

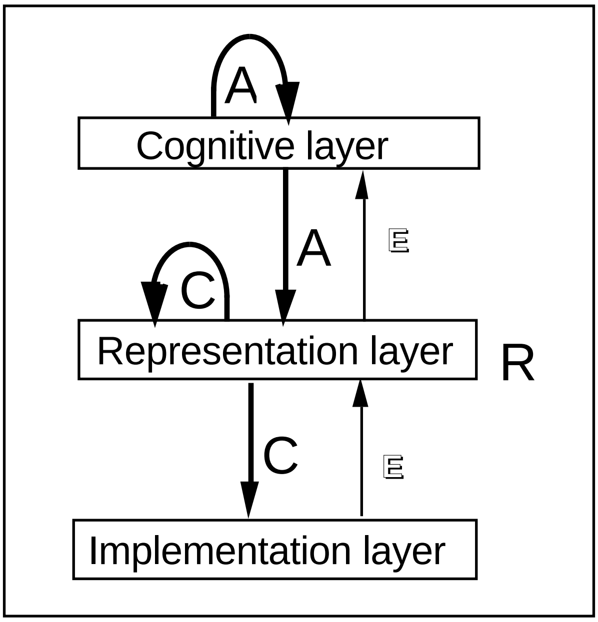

Figure 1: Three levels of knowledge

The arrows labeled by an A indicate knowledge acquisition mappings; E indicates explanation and the C stands for compilation. The term compilation is used here in a very broad sense and includes various kinds of high-level program transformations as well. From the programming point of view in ARC-TEC an expert system shell is under development which supports these aspects. It is as domain specific as necessary in order to be efficient but relies as much as possible on domain independent knowledge. In order to be guided by concrete problems before a large knowledge base is attacked we concentrate first on the task to provide a connection between two specific islands: The automatic generation of a work plan on the basis of CAD data.

The structure of ARC-TEC can be described by the different kinds of models which are used and by their representations. In the terminology of KADS [Bre89] the cognitive layer is structured by the conceptual and the design model. The conceptual model describes informally the expert's reasoning. The still informal design model presents the approach of the system. On the representation layer the design model has its formal counterpart. ARC-TEC's representation language is an extension and modification of KL-ONE [Bra85]. The role of the A-box is played by AFFIRM, which uses rules both in the forward and backward mode; the T-box, TAXON, is used for conceptual hierarchies. In addition we employ a constraint system, CONTAX, utilizing the conceptual hierarchies. These language tools allow a unified development of specialized representation languages for various purposes. We distinguish two types of compilation steps, horizontal ones (on the same language level) and vertical ones, down to lower levels. The lowest level is LISP, which is also the implementation language; presently we are only interested in the algorithmic efficiency of our tools and methods (we regard LISP-to-C translation as a pure software engineering project which should be attacked only after things have been stabilized).

3 Overview of the Present State of ARC-TEC

The levels of figure 1 reflect roughly the organizational structure of ARC-TEC. The acquisition group is responsible for the cognitive layer and the A-arrows. The representation group deals with the representation layer and its ingoing and outgoing arrows. The compilation group is concerned with the implementation layer and the C-arrows. It must be emphasized that there is an intense interconnection between these activities, the integration aspect plays an important role for the whole project.

3.1 Knowledge Acquisition

Knowledge acquisition in the ARC-TEC project is based on a thorough analysis of the expertise in the application domain, i.e. mechanical engineering. The performed analysis included historical, sociological and cognitive aspects and it revealed which traces of expertise are available as information sources for knowledge acquisition and how the target problem is usually solved in practice. The latter results were used to specify a model of expertise [Bre89] and to identify the different types of knowledge which are to be acquired.

Based on these results, an integrated knowledge acquisition method and two elicitation tools are being developed. We will first present some results from the analysis of mechanical engineering knowledge and then we will briefly describe the integrated knowledge acquisition method and the elicitation tools.

3.1.1. Analysis of expertise

The activities of practitioners and theoreticians in mechanical engineering yield different *traces of expertise* which can be utilized for knowledge acquisition:

1) Theoreticians are usually concerned with general rules. The general knowledge which renowned theoreticians accumulated in their research can be found in various *text books*.
2) The specific solutions which practitioners have found over a number of years are stored in filing cabinets or databases of companies. These *records of previously solved cases* constitute an important collection of mechanical engineering knowledge.
3) Through their possibly implicit *expert memories*, which they have acquired over a number of years, practitioners possess an expert classification for the various types of workpieces.

Cognitive analyses have shown that people form libraries of experiences from previously solved cases [e.g. Rie89]. Therefore it is not surprising that adapting previous solutions to newly arising situations (modification planning) is the most frequently used planning procedure in mechanical engineering [Sch90].

The abstract types of processing which are performed in modification planning are described in the model of expertise (or interpretation model in KADS terminology [Bre87]) which can be sketched as follows: From the given geometrical and technological data of the workpiece and description of the available production environment an abstract feature description of the workpiece and an abstract context specification are obtained through the application of abstraction or classification knowledge. With these abstract workpiece and context descriptions a skeletal plan can be associated which may be seen as an abstraction of a concrete production plan. The skeletal plan is then refined with the help of the concrete workpiece and the factory data so that an executable production plan is obtained.

3.1.2. Integrated Knowledge Acquisition Method

The integrated knowledge acquisition method proposes a combined knowledge elicitation from the different information sources which is guided by the model of expertise. In a first step, an informal knowledge base is constructed at the cognitive level. This informal knowledge base is basically a collection of knowledge units and explanation structures. At this informal level, a first verification and documentation of the collected knowledge is performed. In a second step, the informal knowledge is translated from the cognitive to the representation layer and a formal knowledge base is obtained. These two steps roughly correspond to the two arrows labeled "A" in figure 2.

An overview of the acquisition method is presented in Figure 2. It consists of four episodes, which can be flexibly interspersed. Whereas the first three episodes (1. explanation of solved cases, 2. comparison of similar or related knowledge units, 3. competence delineation) concern the elicitation and analysis of knowledge at an informal level, the fourth episode is concerned with the formalization of that knowledge (4. formalization phase). The first three phases will be described in some detail while the formalization phase will be only outlined.

Explanation Episode: In this phase, information which is relatively general and supposedly relevant for the target tasks of the future knowledge based system is selected from one or several appropriate sources. Independently from this selection of general information, a set of prototypical previously solved cases is determined. In the explanation episode the major task for the domain expert consists of applying the selected general information to the previously solved cases by explaining these cases according to the structure of the model of expertise. Through the model-oriented explanation of each case, the completeness of the general information with respect to the specific case is established, and it is assimilated to the model of expertise.

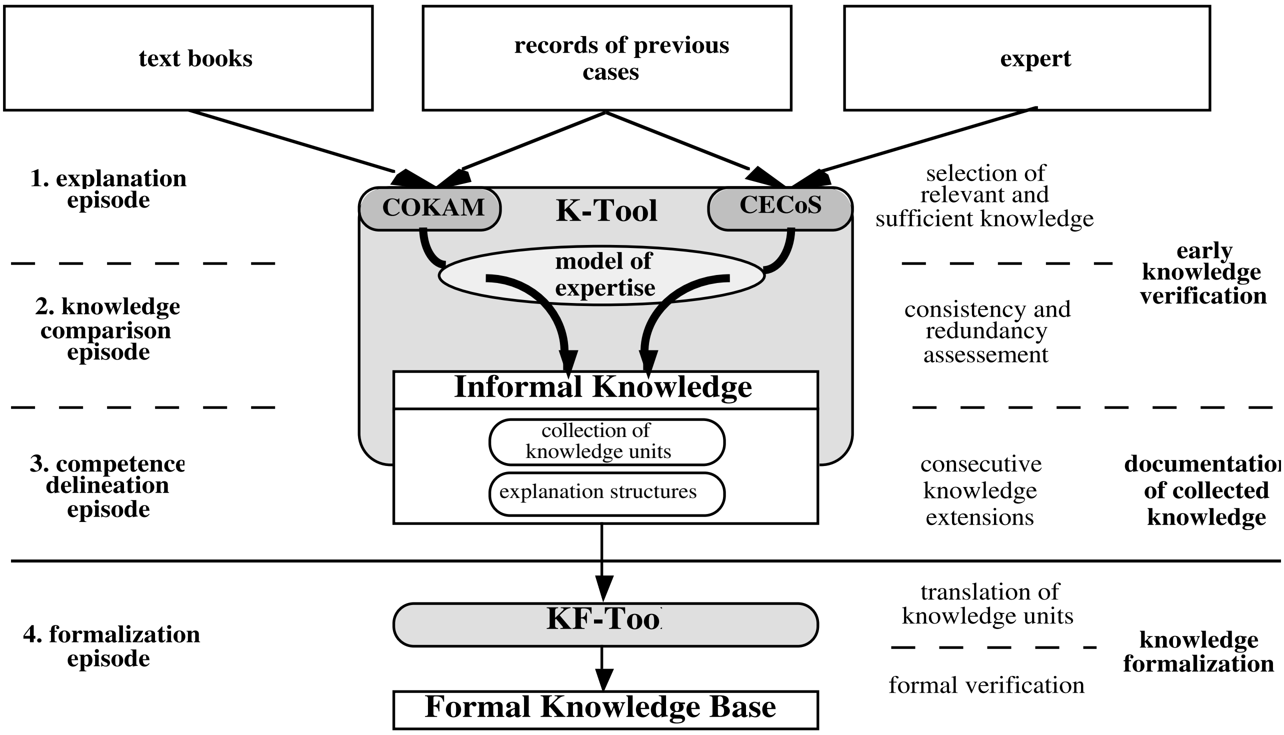

Figure 2: Integrated knowledge acquisition method

Knowledge Comparison Episode: After two or more cases have been explained, the knowledge units in the different categories of the model can be compared to one another. During this comparison, some knowledge units may be found to be redundant, others may be found to be generalizable, still others may require a differentiation. If a contradiction is found, one or several explanations may have to be revised. This may result in an elimination or adjustment of some knowledge units. Each knowledge unit is stated as general as possible and as specific as necessary so that all cases can be explained.

Competence Delineation Episode: In this episode the possible competence which is inherent in the already established (informal) knowledge base is to be delineated. In a most conservative assessment, it is noted that the acquired knowledge is sufficient for successfully solving those problems which were used for knowledge elicitation. The acquired knowledge may in addition be used to solve problems which at some level of generality are structurally identical to the prototypical problems used in the knowledge acquisition phase (e. g. for modification planning). By decomposing the expert´s explanation structures into meaningful segments, solution methods for various subtasks may be identified. By combining solution methods from the different previously treated problems, solutions to structurally new problems may be created. The described competence delineation allows to determine whether iterations of the previously performed knowledge acquisition episodes should be performed or whether the knowledge base already satisfies the requirements of the intended user of the expert system.

Formalization Episode: In the formalization episode, each knowledge unit is translated into its respective formal representation and stored in the formal knowledge base. Thereafter the previously informal explanations are formally examined by traversing the stored explanation structures, this time, however, with the formalized knowledge units. For those explanation structures which can be successfully traversed the early knowledge verification is formally confirmed. Unsuccessful traversals indicate a bug in the execution of the knowledge acquisition, e. g. insufficiencies in the informal knowledge base, errors in the translation from the informal to the formal knowledge units, etc.

Recent research within the KADS-community is developing formal foundations and logical specifications for describing the conceptual model [Wet90]. The approach of KADS II and related work, thus proposes to formalize models relatively soon, while the verification of the acquired knowledge is of less concern. In the current research, on the other hand, a formalization is proposed only after a verification and documentation of the acquired knowledge has been performed. Through early knowledge verification, relevance, sufficiency, redundancy and consistency may already be assessed with informally represented but model-centered knowledge.

3.1.3. Knowledge Elicitation with COKAM and CECoS

For an integrated knowledge acquisition, knowledge elicitation tools [Jac89] are required which select the relevant information from the various traces of expertise. COKAM and CECoS are such knowledge elicitation tools, each performing a joint elicitation from two traces of expertise. COKAM performs a knowledge acquisition from texts, which is enriched by utilizing records of solved cases. With CECoS previously recorded problem solutions are combined with an expert´s high level understanding of the global structure of a task domain. Figure 2 indicates the role of COKAM and CECoS within the proposed integrated knowledge acquisition framework.

In COKAM, an expert first selects texts and text segments which in his opinion contain relevant information for performing the target tasks of the future expert system. Independently of the expert´s selection of text, the knowledge engineer selects previously solved cases from a filing cabinet or a data base. Then the expert explains each case with the help of the selected text segments and his common sense knowledge. The common sense knowledge is thereby used to fill the information gaps in the collection of text segments. The operations and their sequence which are performed by the expert in COKAM are similar to the cognitive processes which a human performs when he studies a text in order to extract specific task knowledge. In particular, those text segments which appear to be relevant, are selected for further processing. They are assimilated to some prior knowledge structure until all slots of the knowledge structure are filled from the text, from hypotheses or through common sense.

In CECoS, a hierarchical classification of problem classes is performed by eliciting global judgements from human experts. For example, after a complete paired comparison of the cases has been performed by the expert, a hierarchical ordering of problem classes may be obtained by a hierarchical cluster analysis. In the application phase, the expert has to generate appropriate feature descriptions for each class, so that he can explain the various class memberships of the cases and the different class subsumptions. Since only the essential characteristics of a previous case and not all the details are stored in human memory, CECoS first obtains an extensional definition of the various production classes with the particular cases used. Thereafter an intensional definition is obtained by having the expert generate appropriate feature descriptions for each class. The knowledge elicitation with CECoS is thus well adjusted to the properties of human memory.

The joint application of COKAM and CECoS allows the integration of two traces of expertise which contain general information with a collection of specific cases. Similar to humans who can learn from general statements and from specific examples by relating the general information to the specifics of the examples, both traces of expertise are used for the purpose of knowledge acquisition. This is accomplished by having the human expert explain the specific examples with the general information. This procedure has been shown to be quite useful for knowledge acquisition [Mar90]. Furthermore, the so acquired knowledge should be well suited for the construction of an expert system which can give good explanations.

3.2 Knowledge Representation

The R-part of ARC-TEC has to provide the higher-level, application-oriented representation languages for the project. To achieve this, the basic tools and languages of the C-part are used to define suitable representation formalisms which then serve as targets of the acquisition efforts of the A-part. Any knowledge represented in these fomalisms can then be subjected to the compilation tools of the C-part. So the R-part plays an integrating role in ARC-TEC; its success depends heavily on the communication with the other parts of the project.

While the problem studied in detail in ARC-TEC is the generation of work plans from CAD data (as this is an example for illustration purposes, only a restricted area is considered), the underlying task of representing an significant portion of every-day knowledge of mechanical engineering is kept in mind. From the R-part's point of view this leads to the identification of several problem categories.

Bearing in mind the life-cycle of an arbitrary ME-product, begining with the design, followed by the generation of the work plan, the manufacturing of the work piece, the assembly of the product and finally the quality checks to certify the correctness of the product and the manufacturing process, it is possible to identify the problem categories and classify them as **diagnosis problems** (e. g. the detection of deviations in the manufacturing process), **configuration problems** (e. g. finding the optimal set of tools for a given NC-machine configuration) and or **planning problems** (obviously: generation of work plans from CAD data) [Pri89] . There is no strict separation between these categories; the design of a product, for example, can probably be described as a blend of configuration and planning.

Any given task in one of these problem categories leads to special requirements concerning the representation of the knowledge necessary to solve the problem, but it is important to observe the huge overlap between the knowledge sets for the different categories: Each task has to deal with knowledge about the product, some of which is important for every possible task (e. g. basic geometric information). Knowledge about the manufacturing environment (shop floor) governs the whole product life cycle.

Therefore the definition of an **integrated knowledge-based product model** (Integriertes Wissensbasiertes Produktmodell **IWP**) is useful. This model allows a description of a workpiece providing all necessary information to perform the tasks mentioned above at any time during a product's life cycle. (See also [Spu86])

The definition of this IWP is the central mission of ARC-TEC part R.

Thinking about a system for the generation of work plans from CAD data along the lines described in 3.1.1, the following representation tasks can be identified:

- work piece representation by geometrical and technological data. Definition of features as higher-level primitives.
- complete and/or partial knowledge about the machining process ("skeletal plan")
- shop floor representation, including machine tools
- dynamic knowledge describing the task to perform ("How to plan")

At the moment a language for the representation of the geometrical/technologial data of a work piece has been completed [Ber90a]. Using this language, the geometry of a work piece is described by combining basic surface primitives. The technological data are represented as attributes of these surface primitives. A transformation of a STEP datafile [Ber90b] into our representation language is possible, forming a bridge to the CAD world.

Given such a representation of a work piece, it is possible to recognize areas of special interest in the work piece. These so-called application features form a higher-level description of the work piece bearing tight relationships to the future manufacturing process. Straight-forward abstractions of the work piece description (e. g. a categorization of its dimensions in relation to the available machine tools) are also considered ("qualitative features").

Currently a preliminary set of features has been defined, the complete feature description language is under development. By using these formalisms, the given CAD representation of a product (given as a STEP datafile) can be transformed into a feature-based representation of the product suitable for the intended task (planning). This concept is illustrated in figure 3.

The feature description named in figure 3 is stored in a data base and is thus available for generative and variation planning. In case of variation planning, a classification of the feature structure can be used to select a complete skeletal (process) plan. In case of generative planning the recognized sequence of features can be used to find parts of skeletal plans or, in the worst case, only single machining operations, which then must be merged to a complete plan. (See also [Fin89])

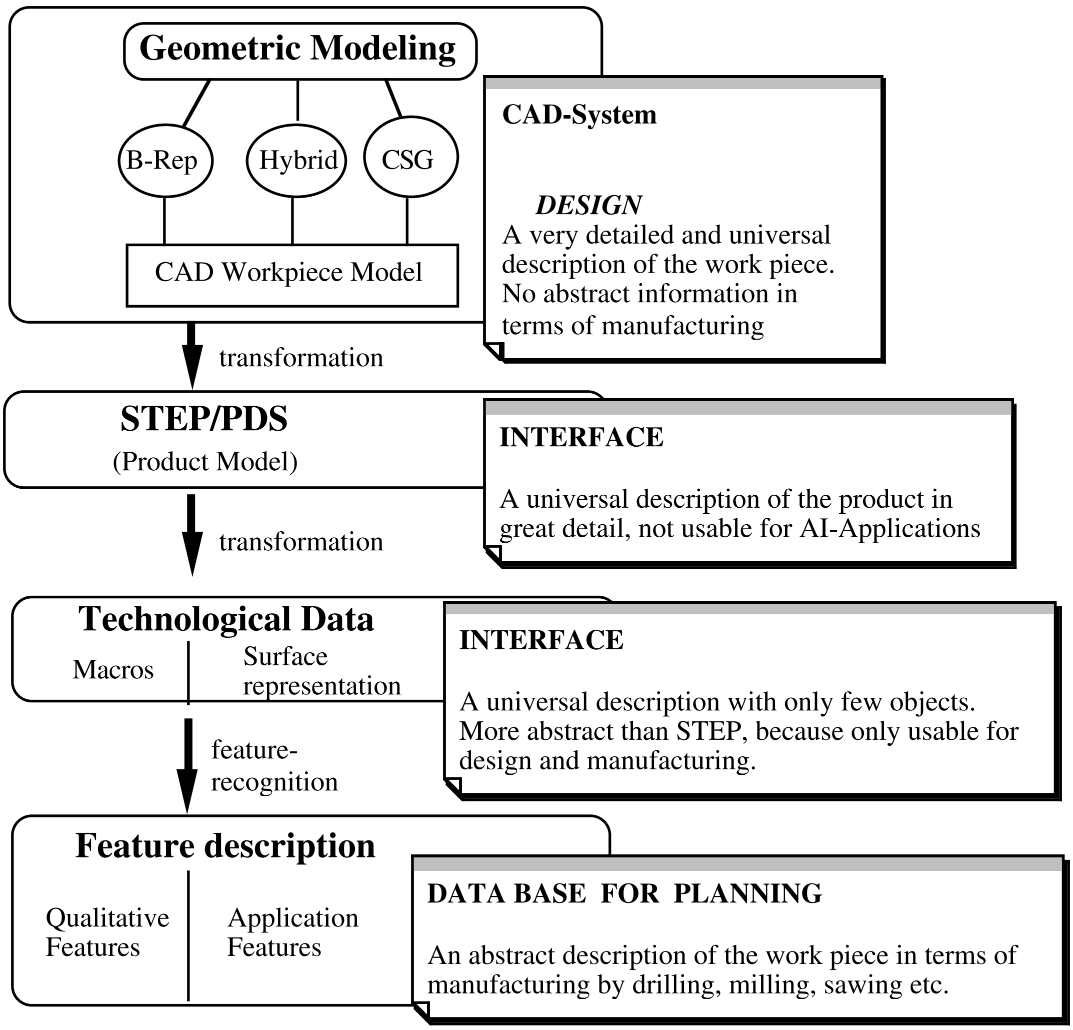

Figure 3: Representation levels

In order to facilitate this process of finding a skeletal plan, the skeletal plans will be hierarchically organized. The feature descriptions of the work pieces form a corresponding hierarchy, as the definition of the features reflects knowledge about the manufacturing process. Consequently, the feature description of a work piece can be used to find the matching skeletal plan [Saf90] .

Our work has shown that the representation of a work piece in terms of geometry and technology is necessary to allow universal generative planning. Although existing feature description languages for design have similarities to our feature descriptions for manufacturing, there are principal differences (e.g. the correlation between manufacturing features and shop floor) which make universal generative planning based only on design features impossible [Cha90].

At the moment we concentrate on the representation of skeletal plans and on the definition of suitable feature description languages. Given a sufficiently sophisticated feature description of work pieces and a matching hierarchy of skeletal plans, together with the knowledge about merging skeletal plans, a very flexible generation of work plans from CAD data should be possible. We hope to prove this in the near future.

3.3 Knowledge Compilation

The compilation group deals with expert system tools and their compilative implementation. In particular, declarative representations are considered. Their high-level description facilitates readability and maintenance of knowledge bases; their orientation toward logic enables clear semantics. However, the processing of large declarative knowledge bases becomes efficient only with the use of modern implementation techniques. The greater distance from hardware imposes high demands on 'intelligent' compilation methods. The term 'compilation' is used here in a broad sense: Besides 'vertical' translation of high-level constructs into constructs of a lower, machine-specific level, also 'horizontal' transformations within one language level (e.g. by partial evaluation) are examined. Following the declarative paradigm, a lot of work can be done horizontally on the source level in preparation of vertical compilation.

Declarative representation formalisms are examined as modular stand-alone tools and are integrated into an extended KL-ONE-like expert system shell. The following main components are considered: backward rules, forward rules, taxonomies, and constraints.

Due to the different characters of the formalisms, various compilation techniques are employed. Horn clauses of the assertional ('A-box') component AFFIRM are executable in forward and backward direction. According to their reasoning direction they are compiled vertically to an extended Warren Abstract Machine, the Relational-Functional Machine (RFM see below). The taxonomical component precomputes implications between concepts in a horizontal compilation step. This helps to avoid unneccessary or redundant inferences at runtime (e.g. by typed unification) and is a useful means for the knowledge engineer to analyze the defined taxonomy. Analogously, an optimized representation of a constraint

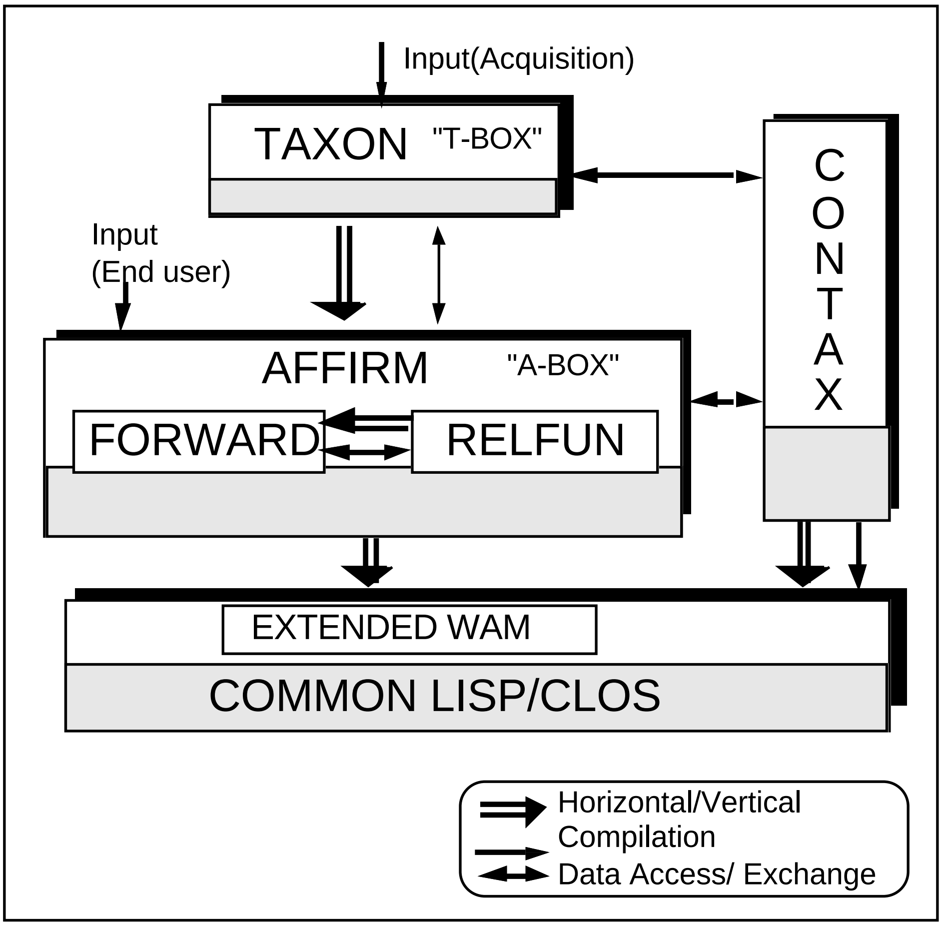

Figure 4: Representation and Compilation Architecture

network can be generated by normalizing transformations. The interactions between these formalisms and their implementation language, LISP, are presented in figure 4.

3.3.1 Backward Rules

The relational/functional language RELFUN and its machine RFM [Bol90] are made available as basic components of AFFIRM. RELFUN amalgamates relations and functions on the basis of 'valued clauses'. These extend horn clauses by special 'foot' premises, which specify values to be returned. Values as well as arguments can be non-ground. Functions can succeed or fail like clauses and enumerate values non-deterministically. Relations act like characteristic functions. They permit functionally nested call-by-value arguments. Higher-order clauses are characterized by a structure or (free) variable in some operator position.

The compilation of this language is performed by various transformation steps at source-code level and introduction of declarative intermediate levels ('classified clauses'). Call-by-value nestings (possibly non-deterministic) are 'flattened', i.e. they are substituted by a new variable to which their value is bound. Higher-order clauses are reduced to 'constant-operator' clauses. RFM code is generated by extending the use of X-registers and 'put'/'get' instructions of the WAM [War83]: a value is put into the register X1 just before a footed clause returns; from there the caller can get it as its argument, as if loaded by a top-level put instruction.

3.3.2 Forward Rules

The forward reasoning part of AFFIRM integrates work from production systems (OPS5, CLIPS [Bro88]), logic programming and deductive databases (bottom-up reasoning [Ull89], [Ban88]). Interpreting horn clauses as implications leads to horn rules. Horn rules provide a good basis for bidirectional reasoning. More general rule structures with disjunctive premises and conjunctive conclusions can be easily transformed into horn rules, too. Although horn logic itself does not prescribe any inference strategy, a kind of top-down reasoning is mostly used in logic programming. We aim at an integrated forward and backward reasoning of RELFUN horn rules. The original rules are horizontally translated into special forward clauses denoting one forward reasoning step. Premises of a forward rule are in turn verified by RELFUN's backward reasoning. Depth-first and breadth-first strategies have been implemented which enumerate the consequences or collect them all at once in a list. For vertical compilation of forward clauses the RFM has been extended by a special retain stack for the derived facts. The subsumption test of a new fact with respect to previously derived ones has been made more efficient by variations of the WAM's unification operations [Hin91].

3.3.3 Taxonomic Reasoning

In general taxonomic formalisms (also called terminological representation languages) based on KL-ONE [Bra85] are used to represent the taxonomical and conceptual knowledge of a particular problem domain on an abstract logical level. To describe this kind of knowledge, one starts with atomic concepts and roles, and defines new concepts using the operations provided by the language. Concepts can be considered as unary predicates which are interpreted as sets of individuals, and roles as binary predicates which are interpreted as binary relations between indivduals.One of the most relevant reasoning services provided by this formalisms is the subsumption service that checks whether one concept is more general than the other [Hol90].

In our technical domain the adequate representation of e.g. geometrical concepts requires to relate points in a coordinate system. For that reason a scheme to extend an abstract taxonomic formalism with a concrete domain has been developed [Baa90]. Examples for concrete domains are the arithmetic of rational or real numbers, temporal or spatial representations or relational databases.

The scheme level already deals with the formal declarative semantics of this formalisms as well as the combination of the respective reasoning algorithms to obtain sound and complete subsumption algorithms. The TAXON system is a first implementation of this scheme extending the abstract language ALC with the concrete domain of real numbers.

3.3.4 Constraint Propagation

Constraints, naively being defined as a set of variables and a relation on them, play a crucial part in many mechanical engineering tasks, e.g. operation scheduling, design, etc. Local relationships are expressed as constraints and are maintained by the attached operations, propagation and relaxation. In CONTAX hierarchically structured domains, representable in the sub-/superconcept formalism of TAXON, are especially considered. Exploiting the structure of the domain while the propagation process takes place [Mac85] as well as relaxation in case of an inconsistent state through generalization in the hierarchy leads to an efficient and declarative constraint handler. Additional features, like explainability and incrementality, require special modules for monitoring the data dependencies [Bac91]. Vice versa, CONTAX can be used to represent the concrete domains of TAXON as constraints.

4 Conclusions

In the ARC-TEC project we have brought together ideas from AI and CIM in a single research environment: AI results such as the KADS knowledge-acquisition methodology, the KL-ONE representation theory and the WAM compilation technology are combined for application to real mechanical engineering problems.

ARC-TEC follows an approach of *intermediate generality*. It is less general than unspecific software tools and less broad than the CYC project. On the other hand, it is more general and broader than specific expert systems for, e.g., planning and diagnosis.

The underlying hypothesis of the enterprise has been that this intermediate level is useful. The results obtained so far point in this direction. Very general tools have turned out to be inefficient; only the concentration on one area (in our case, mechanical engineering) makes things feasible. Avoiding an overly specialized approach enables us both to bridge CIM gaps and to *reuse* models, tools and knowledge bases for many different applications.

In addition, the layered structure from knowledge acquisition, to domain-oriented representation and inference, to various compilation techniques guarantees that the conceptual and software results of ARC-TEC can be applied in the real world.

Bibliography

[Baa90] Baader, F. and Hanschke, P. A Scheme for Integrating Concrete Domains into Concept Languages. Research Report, DFKI Kaiserslautern, Forthcoming, 1990.

[Bac91] Bachmann, B. *HieraCon - A Knowledge Representation System with Typed Hierarchies and Constraints*, Master's thesis, Forthcoming, University of Kaiserslautern, Department of Computer Science, 1991.

[Ban88] Bancilhon, F. and Ramakrishnan, R. Performance Evaluation of Data Intensive Logic Programs. In *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann Publishers, Inc., Minker, J., pp. 441-517, Los Altos, CA, 1988.

[Ber90a] Bernardi, A., Klauck, C., and Legleitner, R. Formalismus zur Repräsentation von Geometrie- und Technologieinformationen als Teil eines Wissensbasierten Produktmodells. Dokument D-90-05, DFKI GmbH, 1990.

[Ber90b] Bernardi, A., Klauck, C., and Legleitner, R. STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch . Dokument D-90-04, DFKI GmbH, 1990.

[Bol90] Boley, H. A relational/functional Language and its Compilation into the WAM. SEKI Report SR-90-05, Universität Kaiserslautern, 1990.

[Bra85] Brachman, R.J. and Schmolze, J.G. An Overview of the KL-ONE Knowledge Representation System. Cognitive Science *9*, 2 (1985), pp. 171-216.

[Bre87] Breuker, J., Wielinga, B., van Someren, M., de Hoog, R., Schreiber, G., de Greef, P., Bredeweg, B., Wielemaker, J., Billaut, J.P., Davoodi, M., and Hayward, S. Model-Driven Knowledge Acquisition: Interpretation Models. Esprit Project 1098, Deliverable Task A1, University of Amsterdam, Social Sciences Informatics, Amsterdam, 1987.

[Bre89] Breuker, J. and Wielinga, B. Models of Expertise in Knowledge Acquisition. In *Topics in Expert System Design, Methodologies and Tools*. North-Holland, Guida, G. and Tasso, C., pp. 265 - 295, Amsterdam, 1989.

[Bro88] Brooke, T.M. COSMIC: CLIPS. AI Expert (April 1988), pp. 71-73.

[Cha90] Chang, T.C. *Expert Process Planning for Manufacturing,* Purude University Assison-Wesley Publishing Company (1990).

[Fin89] Finger, S. and Dixon, J.R. A Review of Research in Mechanical Engineering Design, Part 1 and 2. Research in Engineering Design *1*, 1 (1989), pp. 51-67,121-137.

[Hin91] Hinkelmann, K. Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation. Tech. Rept., DFKI GmbH, forthcoming, 1991.

[Hol90] Hollunder, B., Nutt, W., and Schmidt-Schauß, M. Subsumption Algorithms for Concept Description Languages. In *9th European Conference on Artificial Intelligence (ECAI'90),* 1990, pp. 348-353.

[Jac89] Jacobson, C. and Freiling, M. Knowledge Dependencies. In *Proceedings of the 4TH KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP,* October 1-6 1989, pp. 16-1-16-19.

[Lat78] Latombe, J.C. *Artificial Intelligence and Pattern Recognition in CAD,* North-Holland (1978).

[Len90] Lenat, D.B. and Guha, R.V. *Building Large Knowledge Bases - Representation and Inference in the CYC Project,* Addison-Wesley (1990).

[Mac85] Mackworth, A., Mulder, J., and Havens, W. Hierarchical Arc Consistency: Exploiting Structured Domains in Constraint Satisfaction Problems. Computational Intelligence *1* (1985), pp. 118-126.

[Mar90] Martin, J.D. and Redmond, M. Acquiring Knowledge by Explaining Observed Problem Solving. Knowledge Aquisition: SIGART Newsletter , 108 (April 1990), pp. 77-83.

[Pri89] Pritschow, G., Spur, G., and Weck, M. *Künstliche Intelligenz in der Fertigungstechnik,* , München (1989).

[Ric87] Richter, M.M., Siekmann, J., and Wahlster, W. Eine wissenschaftliche Vision für das deutsche KI-Zentrum. KI , 4 (1987), pp. 30-31, (in German).

[Rie89] Riesbeck, C.K. and Schank, R.C. *Inside case-based reasoning,* Lawrence Earlbaum (1989).

[Saf90] Safier, S.A. and Finger, S. Parsing Features in Solid Geometric Models. In *Proceedings of the European Conference on Artificial Intelligence, ECAI-90,* 1990.

[Sch90] Schmalhofer, F. and Thoben, J. Wiederholungs- Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Teile. Tech. Rept., German Research Center for Artificial Intelligence, Interner Bericht des ARC-TEC-Projektes, 1990.

[Spu86] Spur, G., Krause, F.L., and Armbrust, P. Product models as basis for integrated design and manufacturing. Int. J. Match. Tool Des. Res. *26*, 2 (1986), pp. 171-178.

[Ull89] Ullman, J.D. *Principles of Database and Knowledge-base Systems,* Computer Science Press , Vol. 2(1989).

[War83] Warren, D.H.D. An Abstract Prolog Instruction Set. Technical Note 309, SRI International, Menlo Park, CA, October, 1983.

[Wet90] Wetter, T. First Order Logic Foundation of the KADS Conceptual Model. In *Current Trends in Knowledge Acquisition,* Wielinga, B., Boose, J., Graines, B., Schreiber, G., and van Someren, M., IOS Press, May 1990, pp. 356-375.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekenn-zeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or per anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-92-45
Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46
Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster: WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47
Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48
Bernhard Nebel, Jana Koehler:
Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

RR-92-49
Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:
Heuristic Classification for Automated CAPP
15 pages

RR-92-50
Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51
Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52
Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-53
Werner Stephan, Susanne Biundo:
A New Logical Framework for Deductive Planning
15 pages

RR-92-54
Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55
John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56
Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58
Franz Baader, Bernhard Hollunder:
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59
Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
13 pages

RR-92-60
Karl Schlechta: Defaults, Preorder Semantics and Circumscription
19 pages

RR-93-02
Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist:
Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-03
Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi:
An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

RR-93-04
Christoph Klauck, Johannes Schwagereit:
GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

RR-93-05
Franz Baader, Klaus Schulz: Combination Tech-niques and Decision Problems for Disunification
29 pages

RR-93-06
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: On Skolemization in Constrained Logics
40 pages

RR-93-07
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: Concept Logics with Function Symbols
36 pages

RR-93-08
Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer: COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

RR-93-09
Philipp Hanschke, Jörg Würtz:
Satisfiability of the Smallest Binary Program
8 Seiten

RR-93-10
Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11
Bernhard Nebel, Hans-Juergen Buerckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12
Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13
Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14
Joachim Niehren, Andreas Podelski,Ralf Treinen: Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.): Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf: Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-44
Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

DFKI Technical Memos

TM-91-15
Stefan Busemann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang: Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh:
A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

TM-93-04
Hans-Günther Hein: Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

DFKI Documents

D-92-23
Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24
Jürgen Müller, Donald Steiner (Hrsg.):
Kooperierende Agenten
78 Seiten

D-92-25
Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26
Enno Tolzmann:
Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27
Martin Harm, Knut Hinkelmann, Thomas Labisch: Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28
Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

D-93-01
Philipp Hanschke, Thom Frühwirth:
Terminological Reasoning with Constraint Handling Rules
12 pages

D-93-02
Gabriele Schmidt, Frank Peters, Gernod Laufkötter: User Manual of COKAM+
23 pages

D-93-03
Stephan Busemann, Karin Harbusch(Eds.):
DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

D-93-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1992
194 Seiten

D-93-05
Elisabeth André, Winfried Graf, Jochen Heinsohn, Bernhard Nebel, Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahlster:
PPP: Personalized Plan-Based Presenter
70 pages

D-93-06
Jürgen Müller (Hrsg.):
Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz Saarbrücken 29.-30. April 1993
235 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-07
Klaus-Peter Gores, Rainer Bleisinger:
Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse
53 Seiten

D-93-08
Thomas Kieninger, Rainer Hoch: Ein Generator mit Anfragesystem für strukturierte Wörterbücher zur Unterstützung von Texterkennung und Textanalyse
125 Seiten

D-93-09
Hans-Ulrich Krieger, Ulrich Schäfer:
TDL ExtraLight User's Guide
35 pages

D-93-10
Elizabeth Hinkelman, Markus Vonerden,Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11
Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12
Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-15
Robert Laux: Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-20
Bernhard Herbig:
Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21
Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten